الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التكوين والتعليم المهنيين
Ministère de la Formation et de l'Enseignement Professionnels

Centre National de l'Enseignement Professionnel à Distance (CNEPD)

المركز الوطني للتعليم المهني عن بعد (م.و.ت.م.ب)

COURS DE SYSTÈME D'EXPLOITATION

SÉRIE 02

MÉCANISMES DE BASE D'EXÉCUTION DES PROGRAMMES

OBJECTIF PÉDAGOGIQUE : À la fin de cette série, le stagiaire doit être capable d'appliquer la gestion du matériel dans un système d'exploitation.

PLAN DE LA LEÇON :

INTRODUCTION :

Les ordinateurs modernes se composent de processeurs, de mémoires, d'horloges, de disques, de terminaux, et d'une multitude d'autres périphériques. Dans cette optique, le travail du système d'exploitation consiste à ordonner et à contrôler l'allocation des processeurs, des mémoires et des différents périphériques d'Entrée/Sortie entre les programmes utilisateurs.

Dans cette leçon, nous allons aborder le rôle du système d'exploitation comme gestionnaire du matériel, tout en évoquant les types de difficultés qu'il est amené à résoudre.

I- LA GESTION DU PROCESSEUR (CPU) :

L'un des rôles les plus importants du système d'exploitation, est la gestion de l'unité centrale (CPU). Comme vous le savez, le CPU est le cerveau de l'ordinateur, et donc il est très important pour nous qu'il soit bien entretenu par le système.

Les programmes du système d'exploitation doivent manipuler deux éléments essentiels pour assurer la gestion du CPU, qui sont :

- **Le registre d'état :** Le système d'exploitation doit scruter ce registre pour tirer le maximum d'informations sur l'exécution d'une instruction (erreur, arrivée d'interruption, etc.)
- **Le contexte du processeur PSW :** Ce sont tous les registres utilisés par le CPU pendant l'exécution d'un programme et permettant de l'identifier.

Dans le cas des systèmes multitâches, les programmes s'exécutent à tour de rôle sur le CPU. Le contexte du CPU de chaque programme (tous les registres identifiant ce programme ou **PSW**) est sauvegardé en mémoire et suit le programme pendant sa « vie » (exécution).

II- GESTION DES INTERRUPTIONS :

1- Notion d'interruption :

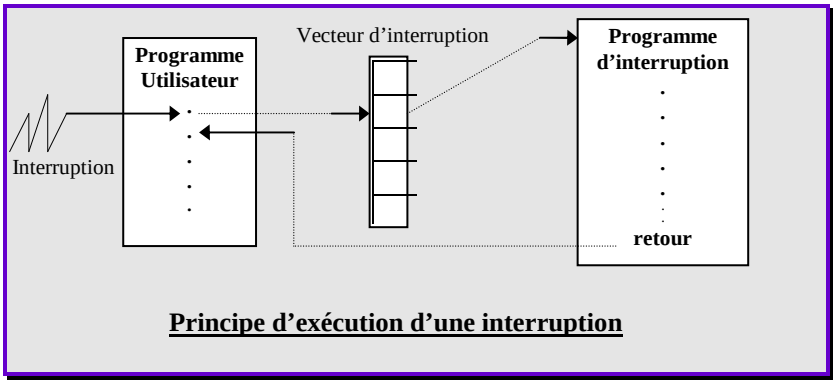
Une interruption est un signal (électronique) qui permet à l'unité centrale de changer d'état. Ce signal apparaît quand certains événements extérieurs au programme se produisent et nécessitent une réaction plus ou moins immédiate du processeur. On réalise ainsi une commutation du contexte du processeur (modification de contexte).

Du point de vue hardware, une interruption est un arrêt technologique qui va provoquer un changement d'état de l'unité centrale et par suite, la suspension du programme en cours d'exécution. Une interruption permet donc de forcer un processeur à suspendre l'exécution d'un programme en cours, et à exécuter un programme prédéfini.

Le programme prédéfini est appelé « programme d'interruption », son adresse de début se trouve en mémoire dans une des entrées du vecteur d'interruptions.

Le programme d'interruption s'exécute dans un contexte distinct de celui du programme interrompu, notamment en ce qui concerne le mode d'exécution, la protection, les informations accessibles, ... etc.

La gestion des différentes interruptions se fait à l'aide d'une logique câblée : « le système d'interruption ». Sur les PC, le contrôleur d'interruptions PIC représente le système d'interruptions. **(Voir cours d'architecture des ordinateurs, structure machine « Série 01 »)**.



Principe d'exécution d'une interruption

Remarque:

A chaque interruption correspond un programme d'interruption bien précis qui s'exécute jusqu'à sa fin puis rend le contrôle du CPU au programme interrompu.

2- Les classes d'interruptions :

Les interruptions sont classées en deux catégories :

- Les interruptions logicielles
- Les interruptions matérielles

■ Les interruptions logicielles :

On parle d'interruption logicielle lorsqu'une interruption est déclenchée par une instruction du microprocesseur et à partir d'un programme.

Exemple :

Dans le cas des microprocesseurs d'INTEL 8086,...,80486 l'instruction assembleur qui effectue une interruption logicielle est « **INT».**

L'instruction **INT 21** : Veut dire au processeur d'aller exécuter le programme d'interruption dont l'adresse de début est stockée au 21^{eme} élément du vecteur d'interruption.

- **Les interruptions matérielles :**

Ces interruptions ne sont pas déclenchées par un programme mais plutôt par des composants électroniques (clavier, imprimante, ...), ce type d'interruptions constitue un mécanisme simple et très efficace pour faire réagir le processeur à des événements déterminés.

Les interruptions matérielles se divisent en deux groupes :

1- Les exceptions ou trappes (internes au CPU) ;
2- Les interruptions matérielles provoquées par des événements extérieurs au CPU.

a- Les exceptions :

Quand un indicateur de changement d'état (bit dans le registre d'état) est modifié par une cause liée à l'exécution d'une instruction en cours, on dit qu'il y'a déroutement ou exception.

Le mécanisme de déroutement a essentiellement pour rôle de traiter une anomalie qui survient dans le déroulement d'une instruction :

Données incorrectes (débordement arithmétique, division par zéro, ...).

- Tentative d'exécution d'une opération interdite par un dispositif de protection (violation de la mémoire, l'exécution d'une instruction privilégiée par un programme non autorisé à le faire).

Chaque numéro de l'exception correspond à une entrée dans le vecteur d'interruptions. Dans cette entrée nous trouvons l'adresse en mémoire du programme d'interruption qui est activé chaque fois que l'exception est déclenchée.

Voici sous forme de tableau quelques numéros d'exceptions du 80486.

N° du vecteur	Fonction	Type
0	erreur de division (division par zéro)	exception
1	exécution pas à pas	interruption
2	interruption non masquable NMI	interruption
3	point d'arrêt	interruption
4	dépassement	exception
5	index invalide	exception
6	code opération invalide	exception
7	coprocesseur non disponible	exception
8	double exception	exception
9		
10	segment d'état de tâche non valide	exception
11	segment non présent	exception
12	débordement de segment de pile	exception

Exceptions et interruptions réservées du 80486

b- Les interruptions matérielles provoquées par des événements extérieurs :

Ces interruptions sont provoquées par des événements extérieurs aléatoires qui requièrent l'attention immédiate du microprocesseur. Celui-ci dispose de deux entrées de demandes d'interruptions qui sont :

- **Interruptions non masquables (NMI) :**

Ce sont les interruptions dont le déclenchement implique l'exécution directe du programme d'interruption correspondant. Donc elles ne peuvent pas être retardées ou masquées par le système d'exploitation.

- **Les interruptions masquables:**

On parle ici, des interruptions qu'on peut masquer, c'est à dire retarder. Ces interruptions peuvent être par la suite démasquées et ainsi le mécanisme d'interruption est réactivé. Sachant qu'on dispose d'une seule entrée pour les interruptions masquables, et afin de multiplier les possibilités du circuit, on fait intervenir un contrôleur d'interruptions **(Voir Série 01 « Structure machine » architecture des ordinateurs)**.

Remarque :

Etant donné le caractère synchrone du déroutement (associé à des instructions en cours) la notion de masquage n'a pas de sens dans ce type d'interruptions. Car comme on peut le deviner, dès que l'anomalie se produit il faut faire rapidement un traitement adéquat. Et donc les déroutements ne peuvent pas être retardés.

Exemple :

Sous MSDOS, Pendant l'impression d'un document, les caractères tapés au clavier ne sont pas pris en compte avant la fin de l'impression. Que s'est-il passé au juste ? En effet, le système **MS-DOS** masque les interruptions qui proviennent du clavier **(interruption numéro 9)** et donc le microprocesseur ignore que des caractères sont tapés au clavier. Dès que l'impression est achevée, MS-DOS démasque les interruptions provenant du clavier, et l'utilisateur voit sur écran les caractères qu'il tape sur le clavier.

3- Le mécanisme d'interruption :

- **Mécanisme de changement d'état :**

Le mécanisme de changement d'état permet en une seule opération indivisible de :

- Ranger dans un emplacement spécifié de la mémoire le contenu courant du mot d'état du processeur (PSW). C'est à dire sauvegarder le contexte du programme interrompu ;

- Charger dans le mot d'état du CPU un nouveau contenu (contexte du programme d'interruption) préparé à l'avance dans un emplacement spécifié de la mémoire.

Ce changement d'état n'est possible que si le processus en cours (ou le programme) se trouve dans un état bien défini, c'est à dire un point observable ou point interruptible.

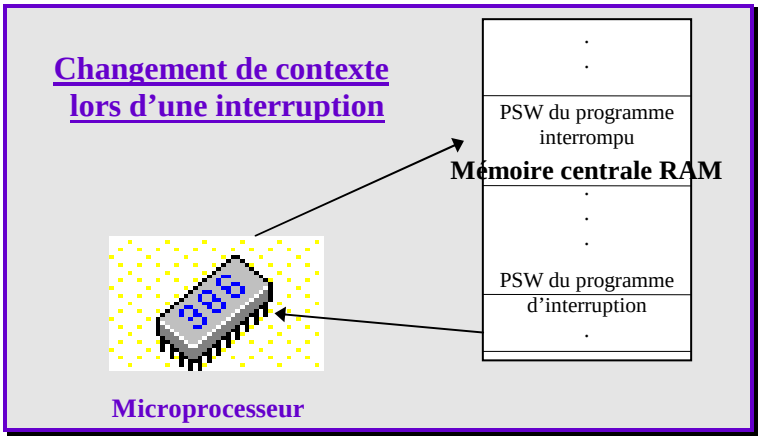
Le changement d'état est déclenché suivant l'état d'un indicateur (bit) consulté par le processeur lors de l'exécution de chaque instruction. Donc, on ne peut pas interrompre un processus dans des points non observables ou non interruptibles.

L'opération de changement d'état d'un processeur est schématisée par la figure suivante :

4- Niveau de priorité :

Un processeur peut être interrompu par diverses causes. Le mécanisme d'interruptions doit pouvoir distinguer ces causes afin de les traiter.

Comme nous l'avons vu, il existe des interruptions masquables qui peuvent être retardées et d'autres qui ne le peuvent pas. Cependant pour assurer leur gestion, on a associé à chaque interruption un degré d'importance, ce qu'on appelle :



Niveau de priorité. Ainsi, si deux interruptions arrivent en même temps, la plus prioritaire passe la première. Pour réaliser ce principe de priorité, deux schémas sont employés :

- Un indicateur distinct est associé à chaque cause d'interruption (on parle souvent de niveau d'interruptions), à chaque niveau est associé un couple distinct d'emplacements pour la sauvegarde et le chargement du mot d'état, en d'autres termes à chaque niveau correspond un programme de traitement distinct, automatiquement activé par le mécanisme d'interruption.
- Un indicateur unique est utilisé pour toutes les interruptions avec un couple unique d'emplacements pour la sauvegarde et le changement du mot d'état. Une information supplémentaire (code d'interruption) contenu dans le mot d'état ou dans un emplacement conventionnel en mémoire permet de distinguer entre les causes possibles des interruptions. La première tâche du programme est de consulter ce code pour déterminer l'origine de l'interruption et appeler la routine appropriée.

Remarque :

En pratique, on trouvera souvent une combinaison de ces deux mécanismes

III- LA GESTION DES ENTRÉES /SORTIES :

Le contrôle des périphériques d'Entrées/Sorties de l'ordinateur est une des fonctions primordiales d'un système d'exploitation. Le système d'exploitation doit envoyer les commandes aux périphériques, intercepter les interruptions provenant des périphériques d'E/S, récupérer les données lues et traiter les erreurs.

Les principes du point de vue matériel d'une E/S sont présentés dans les modules structure des ordinateurs et réseaux. Dans ce paragraphe, nous verrons l'Entrée/Sortie du point de vue software c'est à dire comme partie intégrante du système d'exploitation.

Avant tout, donnons la définition d'une opération d'Entrée Sortie :

On désigne sous le terme général d'entrée/Sortie (**E/S** ou **I/O**) tout transfert d'informations depuis ou vers l'ensemble mémoire adressable. Les E/S se divisent en deux types :

- Les E/S de type caractère ;
- Les E/S de type bloc

Le rôle primordial des programmes du système d'exploitation s'occupant de la gestion des E/S est d'acheminer les données entre les programmes des utilisateurs et les périphériques.

En partant du principe que seul le système d'exploitation connaît les spécificités de la machine et que lui seul peut savoir où se trouvent les programmes des utilisateurs en mémoire, nous pouvons déduire que le système d'exploitation est le plus habilité à faire ce transfert entre programmes utilisateurs et périphériques d'Entrée/Sortie.

Ainsi, pour l'utilisateur chaque port d'E/S ou périphérique a un nom logique tel que LPT1 : pour l'imprimante par exemple. Le système doit faire le lien entre ce nom logique et l'adresse physique (espace d'adressage I/O) d'une façon transparente à l'utilisateur.

De la même façon, les zones de données de l'utilisateur sont identifiées par des noms de variables. Le système doit faire le lien entre ces noms de variables et l'adresse physique en mémoire et pouvoir ainsi effectuer le transfert.

1- Les E/S de type caractère :

Les différentes fonctions du système d'exploitation pour l'entrée et la sortie de caractères permettent de commander des périphériques de type caractère tels que le clavier, l'écran, l'imprimante et l'interface série.

Deux modes de fonctionnement existent pour ce type :

- **Lecture indirecte avec contrôle (COOKED) :**

Elle consiste à contrôler les caractères transférés entre le programme utilisateur et les périphériques. Sur le clavier, les touches de contrôle tel que **CTRL-Z**, **CTRL-C** sont alors reconnues et une routine correspondante est déclenchée.

Exemple 1 :

Quand ce mode est actif et qu'on tape **CTRL-C** sur le clavier, le programme en cours d'exécution est arrêté et l'exécution est retournée au **DOS** (système d'exploitation).

Exemple 2 :

L'instruction READ en pascal permet de lire des données. Ceux-ci ne sont transférés au programme utilisateur (pascal) que si on tape la touche ''entrée''. Dès que le système d'exploitation reconnaît la touche ''entrée'' il prend le contenu du buffer (tampon du clavier) et le transfert vers la zone de données du programme utilisateur.

- **Mode direct (RAW) :**

Dans ce mode dès qu'une touche est tapée elle est immédiatement transférée vers le programme qui en a besoin sans vérifier sa valeur, ni attendre une validation par la touche ''Entrée''.

2- Les E/S de type bloc :

Les périphériques de type bloc mémorisent les données dans des blocs de taille fixe, chaque bloc ayant une adresse propre. Les tailles courantes des blocs vont de 128 octets à 1024 octets. La propriété fondamentale de ces périphériques est qu'ils permettent de lire ou d'écrire un bloc indépendamment de tous les autres. En d'autres termes, un programme peut écrire ou lire n'importe quel bloc stocké dans ces périphériques. Les disques, CD-ROM, Bandes magnétiques,… etc. font partie de ce type de périphériques.

3- Les étapes de déroulement d'une E/S :

L'E/S est une opération très courante dans la vie des logiciels, elle met en jeu plusieurs acteurs en plus du programme utilisateur voulant lire ou écrire sur un périphérique d'E/S. Une opération d'E/S passe par plusieurs phases ou couches avant de s'achever. Les quatre principales couches sont :

- La couche de bas niveau ;
- La couche des pilotes de périphériques ;
- La couche des routines de gestion d'E/S ;
- La couche d'interface avec le programme utilisateur.

Cette Présentation en forme de couches a plusieurs avantages dont le plus important est l'indépendance (dissociation) entre les procédures qui touchent au matériel (qui sont très spécifiques à la marque de la machine) et les procédures de gestion logique des E/S (ne touchent pas au matériel). Ceci est schématisé par la figure suivante :

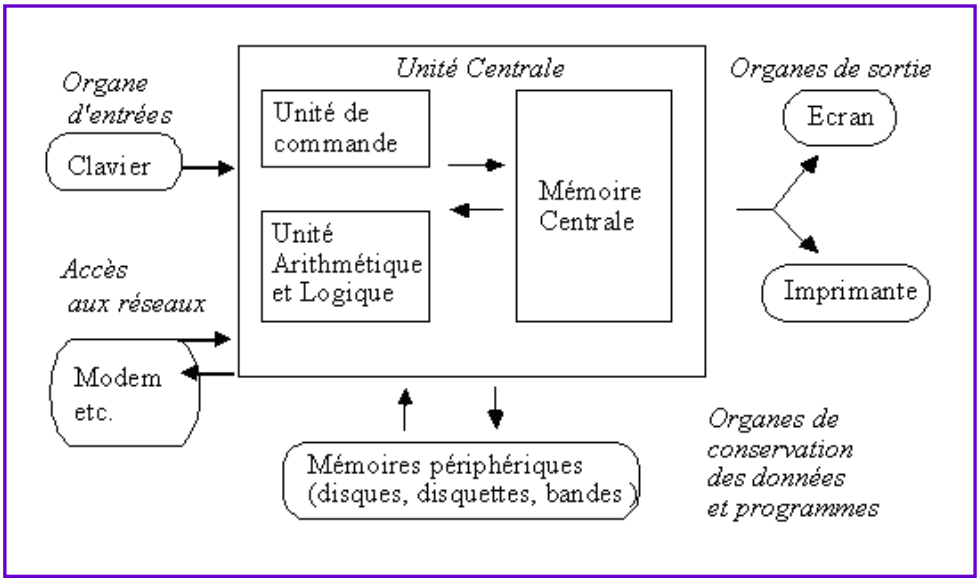


Schéma synoptique d'une entrée/sortie

- **Les routines de bas niveau :**

La couche de bas niveau est très proche du matériel et s'occupe d'effectuer physiquement le transfert et d'activer les circuits associés au périphérique (DMA et contrôleurs). Elle utilise le mécanisme d'interruptions pour communiquer avec les périphériques, et stocke les données devant être acheminées à partir ou vers le périphérique dans des tampons intermédiaires (Buffers d'Entrées sorties).

Exemple1 :

Sur les machines PC ces routines sont appelées BIOS (Basic **I**nput **O**utput System), elles sont écrites par le constructeur du micro-ordinateur, et sont appelées par MS-DOS et les autres logiciels.

Exemple 2:

Sur micro-ordinateur, la lecture à partir du clavier se fait d'une manière très simple :

Les étapes de déroulement de l'entrée de caractères se fait comme suit :

- Chaque fois qu'un caractère est tapé sur le clavier, une interruption matérielle N° 9 (INT 9) est déclenchée ;
- Elle réveille la routine de bas niveau (BIOS) qui se chargera de transférer le caractère tapé du tampon du clavier vers une zone en mémoire centrale ;
- Quand un programme veut lire à partir du clavier, il appelle le système MS-DOS à l'aide de l'interruption logicielle N°21h (l'instruction INT 21h), et exécute ainsi le programme de gestion de l'E/S ;
- MS-DOS de son côté appelle le pilote qui gère le clavier et qui s'appelle CON : et lui demande d'aller chercher un caractère ;
- Le pilote va dans la zone mémoire et cherche le caractère qui a été tapé à l'aide de la routine d'interruption logicielle N° 16h.

- **Les pilotes de périphériques :**

Les pilotes de périphériques sont des programmes du système d'exploitation qui doivent formuler la demande d'E/S sous forme de commandes suivant la nature de l'opération et le type du périphérique. Ils communiquent avec le matériel d'une façon directe ou à travers les routines de bas niveau.

Dans le cas des systèmes multitâches, les pilotes ont pour rôle de gérer les files d'attentes des demandes d'E/S sur un périphérique donné. Ils gèrent également les buffers et tampons (mémoires de stockage intermédiaire entre le périphérique et le programme utilisateur).

Notons que plusieurs logiciels s'en passent du système d'exploitation pour effectuer leurs opérations d'E/S et donc accèdent directement aux composants et aux ports en utilisant leurs propres pilotes (DRIVERS).

- **Les routines de système de gestion d'E/S :**

C'est un ensemble de routines de haut niveau : Elles s'occupent de la gestion logique de l'opération d'E/S et de la prise en compte de la requête (demande) d'E/S du programme utilisateur. Parmi ses tâches élémentaires nous citons :

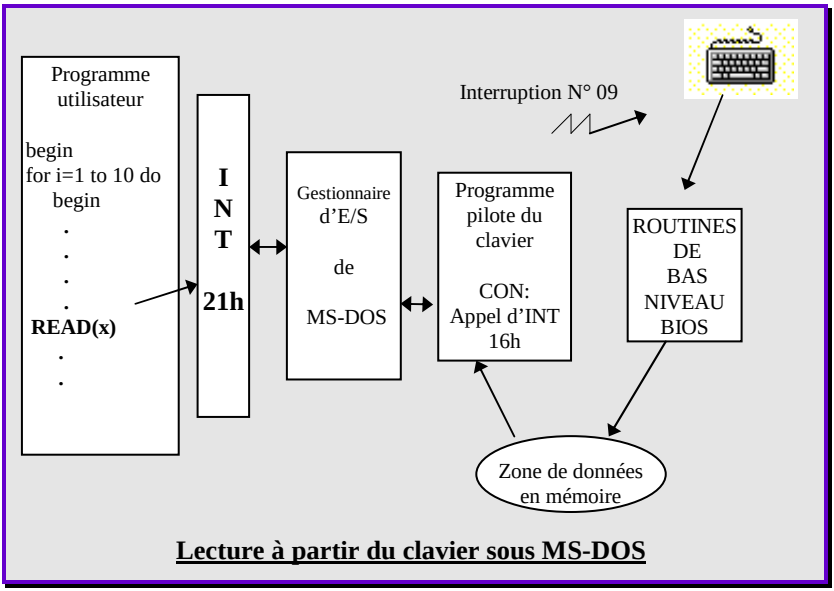
- Correspondance entre le nom logique du périphérique et de son adresse physique (port d'E/S) ;
- Vérification des droits d'accès et établissement des modes d'accès ;
- Allocation de mémoire intermédiaire (tampons) nécessaire à cet effet ;
- Acheminer les données lues à partir du périphérique dans la zone de données du programme utilisateur appelant ;
- En cas d'erreurs d'E/S prévenir le programme utilisateur et afficher un message.

- **L'interface de la mémoire :**

L'interface d'appels au système (System call) : Elle comprend les fonctions et programmes qui jouent le rôle d'interface entre le programme utilisateur (client) et le système d'exploitation (serveur) devant effectuer l'opération. Elles se présentent sous plusieurs formes :

- Fonctions d'appel aux bibliothèques système, accessibles à travers les langages de programmation tel que PASCAL, FORTRAN, etc... .
- Interruptions logicielles réservées.

V- GESTION DE LA MÉMOIRE :



Lecture à partir du clavier sous MS-DOS

La mémoire est une ressource de stockage de données et de programmes. Elle est subdivisée en une suite finie de composants appelés emplacements. La mémoire est une ressource nécessaire à l'exécution d'un processus ou programme. Ce dernier ne peut s'exécuter sans être chargé (au moins partiellement) en mémoire centrale (RAM).

Objectifs :

Le gestionnaire de la mémoire est un module du système d'exploitation dont le rôle est la gestion de la mémoire principale. Il doit viser les objectifs suivants :

- **L'allocation :** Tout programme qui s'exécute en mémoire a besoin d'espace mémoire (RAM) pour stocker ses données et variables. Le gestionnaire de mémoire, est un ensemble de fonctions et procédures du système d'exploitation qui s'occupe de cette allocation.
- **La réallocation :** Un programme ne connaît pas à priori l'environnement de son exécution. Lorsqu'un programme terminé libère de l'espace mémoire, il est nécessaire de réorganiser les programmes en mémoire afin d'optimiser l'utilisation de cette ressource.
- **La protection :** La coexistence de plusieurs processus en mémoire centrale nécessite la protection de chaque espace mémoire vis à vis des autres. Ainsi, un processus $\mathbf{P_1}$ ne peut accéder à l'espace d'un processus $\mathbf{P_2}$ que si ce dernier l'autorise.
- **Le partage :** Parfois, il est utile de partager un espace mémoire entre plusieurs processus. Ainsi, le sous-système de gestion de la mémoire doit autoriser des accès contrôlés sans compromettre la protection.

Exemple :

Un éditeur de texte peut être partagé entre plusieurs utilisateurs d'un système à temps partagé.

Il existe plusieurs stratégies d'allocation de la mémoire. Nous citons parmi elles la **PAGINATION** et la **SEGMENTATION**.

1-Segmentation de la mémoire :

Comme nous l'avons vu pour la gestion de la mémoire virtuelle, les programmes et les données doivent être divisés en parties pour faciliter leur manipulation.

La segmentation est une façon de diviser le programme et les données, suivant leur signification logique par exemple un programme peut être divisé de la façon suivante :

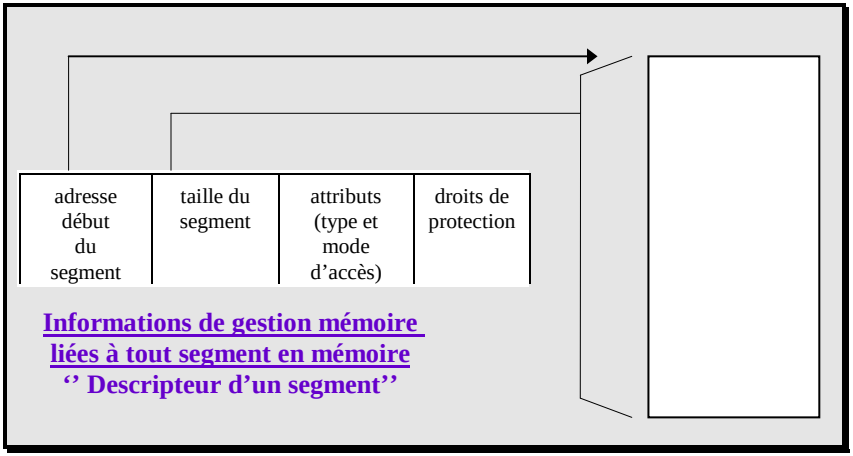
- Le premier segment comporte les données et les variables ;
- Le deuxième segment contient le programme principal ;
- Le troisième segment contient les sous programmes ;
- Le quatrième segment contient la pile.

Le système d'exploitation ne gère plus des cases mémoires mais des segments, il les mets en attente sur le disque quand on n'a pas besoin d'eux et les charge en mémoire dès qu'ils deviennent utiles pour l'exécution du programme.

L'autre avantage de l'adressage segmenté est que les adresses générées par les programmes ne voient pas la mémoire physique mais plutôt un ensemble de segments avec des déplacements à l'intérieur de chaque segment ce qui permet :

- Un espace d'adressage plus grand ;
- Une indépendance de l'emplacement des programmes en mémoire (adresses virtuelles).

2-Technique d'OVERLAY (recouvrement) :



Il y a de nombreuses années, les programmes étaient trop volumineux pour entrer dans la mémoire disponible (la taille de la RAM était beaucoup plus petite que la taille du programme). La solution générale adoptée fut de diviser le programme en plusieurs « morceaux » appelés OVERLAYS. Ainsi, pour exécuter un programme volumineux, il suffisait exécuter d'abord son premier overlay. Quand celui-ci est terminé, il est effacé de la mémoire et l'overlay suivant est chargé en mémoire (à partir du disque), à la même place qu'occupait le précédant overlay. Ce mode est effectué autant de fois qu'il y a d'overlays dans un programme.

L'inconvénient majeur de cette méthode de gestion de la mémoire se résume en deux points :

- La division d'un programme en overlays doit être faite par le programmeur lui-même. Ce qui représente un long et fastidieux travail, car le programmeur doit gérer lui-même son espace de travail en mémoire ;
- Si le programmeur n'est pas expérimenté, ses programmes risquent de devenir très lents, dans le cas où le programme effectue plusieurs va et vient entre le disque et la mémoire (chargement et déchargements d'overlays).

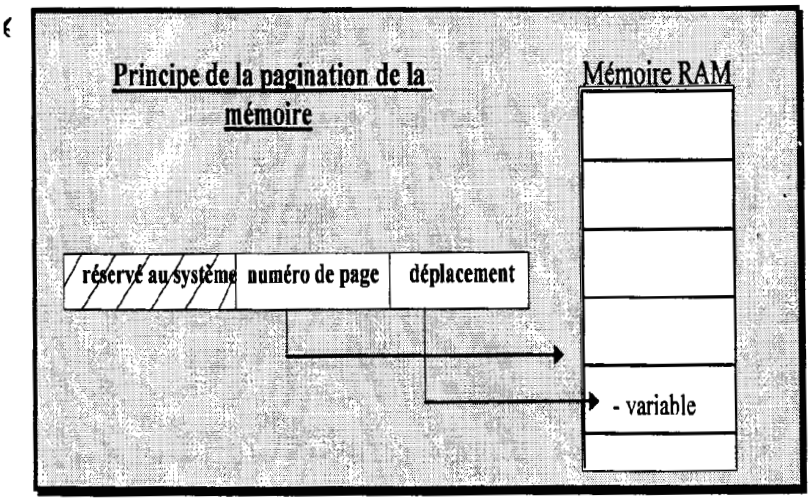
La finalité de ce type de fonctionnement est que l'utilisateur doit gérer lui-même sa mémoire centrale (RAM).

3-Mémoire paginée :

La notion de pagination de la mémoire est une technique qui est venue combler les insuffisances de la segmentation de la mémoire et la complémenter.

Le principe est très simple, il consiste à diviser la mémoire RAM en plusieurs blocs de même taille (par exemple de 4 K octets). Ainsi, un programme voulant accéder à une donnée en mémoire doit lui accéder à travers la page qui la contient et non pas directement. Pour cela, l'adresse est composée de deux informations essentielles qui sont :

- Le numéro de la page ;
- La position à l'intérieur de la page.

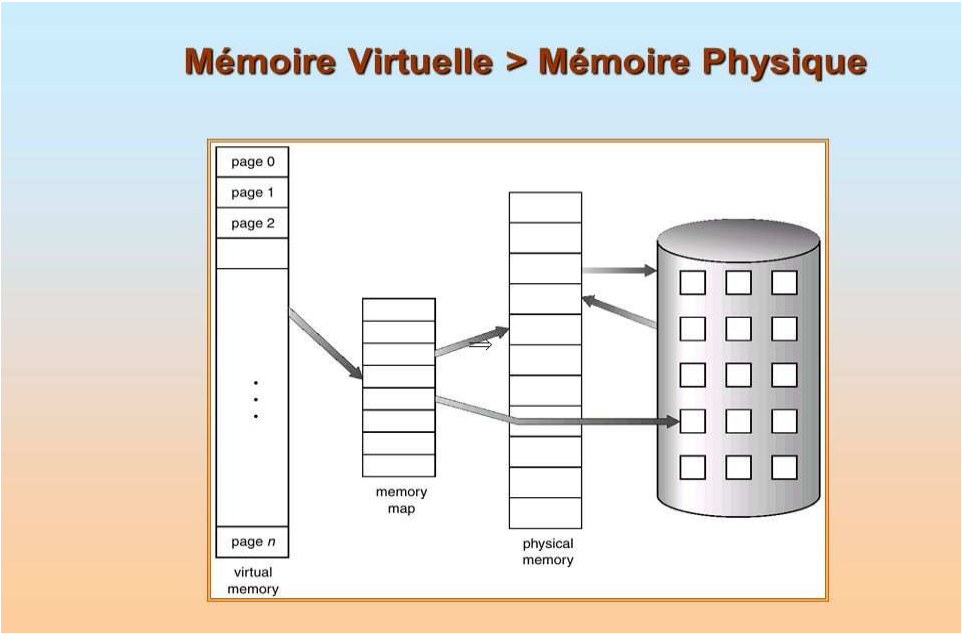


4-Mémoire virtuelle et SWAPPING :

Pour pallier aux inconvénients du système d'overlays, la notion de mémoire virtuelle a vu le jour. Le but est de décharger l'utilisateur de gérer la mémoire, et de réserver cette tâche de gestion des longs programmes au système d'exploitation.

L'idée de base de la mémoire virtuelle est que la taille du programme, des données et de la pile peut dépasser la mémoire disponible. Le système d'exploitation garde en mémoire les parties de programme qui sont utilisées et stocke le reste sur le disque dur.

Dès que le programme tente d'exécuter une instruction (ou accéder à une donnée) qui n'est pas chargée en mémoire, mais qui se trouve sur le disque dur, une interruption de type exception se produit et le système d'exploitation est réveillé. Il charge alors la partie qui contient cette instruction (ou cette donnée) et rend le contrôle au programme interrompu de l'utilisateur.



RÉSUMÉ :

Dans cette leçon nous avons exposé, du mieux qu'on a pu, les plus importantes fonctionnalités de gestion du matériel par le système d'exploitation. Les plus importantes sont :

- **Gestion du CPU**: Le système d'exploitation, doit à tout moment connaître l'état du processeur et pouvoir le partager à travers plusieurs programmes utilisateurs en sauvegardant le contexte du CPU.
- **Les interruptions** Etant très importantes, le système d'exploitation doit pouvoir les gérer en préparant un ensemble de programmes d'interruption, et en sauvegardant les contextes des programmes interrompus. Il peut gérer deux types d'interruptions :

  - Les interruptions matérielles ;
  - Les interruptions logicielles.

- **Les opérations d'E/S** Sont également prises en compte par le système d'exploitation, plusieurs étapes sont définies à partir du moment où la requête est présentée jusqu'à la réalisation matérielle de l'E/S à l'aides des routines matérielles en passant par les pilotes de périphériques et du gestionnaire d'E/S.
- Enfin, **la gestion de la mémoire** reste une tâche importante et ardue du système d'exploitation. Tous les types de gestion ont été présentés : la segmentation, le recouvrement, la pagination, la mémoire virtuelle.

La gestion du matériel par le système d'exploitation étant présentée, nous pouvons désormais voir la gestion logique du logiciel, des données et des utilisateurs. Ceci fera l'objet de la prochaine leçon.

EXERCICES D'APPLICATION :

Question 1:

Quel est le rôle d'une exception ? Dans le 80486 quel est le numéro de l'exception qui est déclenchée chaque fois qu'un programme essaie d'accéder à un segment non présent en mémoire ? Que se passe-t-il ensuite ?

Question 2 :

Classer par ordre croissant de priorité les interruptions suivantes :

- Exceptions ;
- Interruptions logicielles ;
- Interruptions matérielle non masquables ;
- Interruptions matérielles masquables.

Question 3 :

Quel est le rôle du pilote de périphériques dans un système d'exploitation ?

Question 4 :

Pour accéder à un disque dur, le système d'exploitation utilise des buffers ou tampons. Quel est le rôle de ces buffers ?

Question 5 :

Citer les objectifs d'un gestionnaire de mémoire ?

Question 6 :

Répondez par vrai ou faux ?

- Les segments de mémoire sont de taille fixe ;
- Une page ne peut contenir en mêmes temps des données et des instructions ;
- Un segment peut contenir en mêmes temps des données et des instructions ;
- La mémoire virtuelle ne s'applique pas aux données mais seulement aux programmes ;
- Le programme d'interruption s'exécute dans un contexte distinct de celui du programme interrompu.

CORRECTION DES EXERCICES :

Question 1 :

1- Une exception a essentiellement pour rôle de traiter une anomalie dans le déroulement d'une instruction machine :

- Données incorrectes (débordement arithmétique, division par zéro, ...) ;
- Tentative d'exécution d'une opération interdite par un dispositif de protection (violation de la mémoire, l'exécution d'une instruction privilégiée en mode esclave)

2- Le numéro de l'exception est le 11.
Un programme d'interruption est alors exécuter, généralement c'est le gestionnaire de la MEMOIRE VIRTUELLE, son rôle est de continuer l'exécution du programme interrompu, il doit:

- Chercher une place libre en mémoire ;
- Réserver cette place mémoire ;
- Localiser sur le disque dur le segment absent de la mémoire ;
- Charger ce segment en mémoire à la place réservée ;
- Exécuter l'instruction qui a provoqué l'exception.

Question 2 :

L'ordre est le suivant (du moins prioritaire au plus prioritaire) :

1- Interruptions logicielles ;
2- Interruptions matérielles masquables ;
3- Interruptions matérielles non masquables ;
4- Exceptions.

Question 3 :

Les pilotes de périphériques ont pour rôle:

- Formuler la demande d'E/S sous forme de commandes suivant la nature de l'opération et le type du périphérique ;
- Communiquer avec le matériel d'une façon directe ou à travers les routines de bas niveau ;
- Dans le cas des systèmes multitâches, gérer les files d'attentes des demandes d'E/S derrière un périphérique donné ;
- Gérer également les buffers et tampons (mémoires de stockage intermédiaire entre le périphérique et le programme utilisateur).

Question 4 :

Les buffers ou tampons d'E/S ont deux rôles fondamentaux :

- Ils servent de zones de stockage intermédiaires entre les programmes utilisateur et les périphériques. Car, comme vous l'avez vu, ces zones se trouvent entre les pilotes de périphériques et les routines de bas niveau. Un mode de fonctionnement de serveur/client ou de boite aux lettres est constaté ;
- Ils permettent d'accélérer la vitesse de travail. En effet quand un programme écrit dans le disque, le système d'exploitation n'effectue pas l'opération directement dans le disque (qui sont très lents) mais écrit dans ces buffers en mémoire (opération très rapide). De temps en temps seulement un rafraîchissement est établi (transfert des contenus des tampons vers leurs emplacements dans le disque dur). L'avantage est donc d'éviter d'accéder souvent au disque dur, et donc d'améliorer la vitesse des systèmes.

Question 5 :

Le programme de gestion de la mémoire qui est une partie du système d'exploitation doit viser les objectifs suivants :

1- L'allocation de mémoire libre pour les programmes et leurs données ;

2- La réallocation : un programme ne connaît pas à priori l'environnement d'exécution de son programme ;

3- La protection : la coexistence de plusieurs processus en mémoire centrale nécessite la protection de chaque espace mémoire vis à vis des autres ;

4- Le partage : Parfois, il est utile de partager un espace mémoire entre plusieurs programmes comme par exemple les buffers d'E/S qui doivent être utilisés en même temps par les pilotes de périphériques et les routines de bas niveau.

Question 6 :

- Les segments de mémoire sont de taille fixe**. FAUX**
- Une page ne peut contenir en mêmes temps des données et des instructions. **FAUX**
- Un segment ne peut contenir en mêmes temps des données et des instructions. **FAUX**
- La mémoire virtuelle ne s'applique pas aux données mais seulement aux programmes. **FAUX**
- Le programme d'interruption s'exécute dans un contexte distinct de celui du programme interrompu. **VRAI**